Contents

3 GEM RCS Reference

Figures

Tables

Section 1

Introduction to GEM RCS

GEM^{TM} Resource Construction Set (RCS) is a GEM application you use to create resources (menus, dialog boxes, alert boxes, etc.) for your GEM application programs. GEM RCS also lets you incorporate icons and bit-images you create with GEM IconEdit into your resource files.

You don't have to be a programmer to use GEM RCS. You can create all of the application's resources and then give the file to a programmer to include with the program code.

This document is divided into the following sections:

- Section 1, **Introduction to GEM RCS**, explains what a resource is, describes the GEM RCS screen, discusses the mouse techniques you use with GEM RCS, and describes the GEM RCS workspace and how GEM RCS uses memory.
- Section 2, **GEM RCS Tutorial**, acquaints you with basic terminology and techniques as you construct a menu and a dialog box.
- Section 3, **GEM RCS Reference**, contains detailed descriptions of GEM RCS's features, including the parts box, the objects you can use in your resources, the clipboard, toolkit, and menus.
- Section 4, **RSCREATE and Other Technical Information**, describes RSCREATE, a utility program with which you can modify and port resource files. The section also describes how to chain resource files and how to deal with unknown object tree types, and provides some tips for speedy and efficient use of GEM RCS.

Many of the techniques you use with GEM RCS, such as choosing a command from a drop-down menu, are the same as those you use with the GEM Desktop^{TM}. Be sure you have read your GEM Desktop guide and understand these basic operations before you start working with GEM RCS.

WHAT ARE RESOURCES?

In GEM applications, resources are things that appear on the screen (like menus, dialogs, or alerts) that are not actually part of the program code. Instead, resources are kept in a separate resource file, an arrangement that has several advantages.

- As noted previously, the resource file can be created by a non-programmer.
- The resource file can be modified or updated (again by a non-programmer) often without the application code having to be recompiled.
- An application can exist in different national "editions," using the same program code and different resource files for each nationality.
- An application can operate in different machine environments, using resources generated from the same source code.

Resources are made up of objects. "Object," in this sense, is a technical term referring to a specific set of images that can appear on the screen, including empty boxes, boxes containing text, text strings, and the like.

To create a resource--a menu, for example--you combine objects to form an "object tree." The relationship between the objects in the tree is described in family terms: the first object is the "parent"; the objects contained within the parent are the "children."

For a complete description of objects and object trees, see Section 6, "Object Library," in the GEM Application Environment Services Reference Guide. (The GEM Application Environment Services are referred to as GEM AES.)

Using GEM RCS, you can create the following kinds of object trees:

MENU The drop-down menus characteristic of GEM applications are contained in the resource file. No menu can be more than one-fourth the size of the screen. This makes it possible for the part of the screen covered by the menu to be written to a special buffer (see the discussion of the menu/alert buffer in the Introduction to

GEM Programming) and for the screen to be redrawn by the GEM AES from this buffer. A screen redraw from the menu/alert buffer is faster than a redraw handled by the application.

DIALOG To the end-user, a dialog is a GEM application's means of providing or getting information, but GEM applications can use dialogs for other purposes. For example, the GEM RCS toolkit (see Figure 1-1) is actually a DIALOG in the resource file.

Dialogs do not use the menu/alert buffer and thus can be larger than one-fourth the size of the screen. The objects in a dialog "snap" to character boundaries on the screen; this snap makes it easier to align the elements of the dialog.

PANEL A panel is similar to a dialog, except that it does not have the automatic character boundary snap. A panel can be used for anything that requires precise positioning of objects. For example, the GEM RCS parts boxes (see Figure 1-1) are panels in the resource file.

ALERT Alerts (a specialized subset of dialogs) are a GEM application's means of displaying notices, warnings, or error messages. Alerts have a fixed format and a size limit of no more than one-fourth the size of the screen. (They use the menu/alert buffer mentioned previously.) The alert format includes an icon, a 200-character ASCII text string (five lines of no more than forty characters each), and up to three exit buttons. Each exit button has a twenty-character maximum.

FREE This category includes strings and bit-images you want to include in the application, but not as part of the code. In this way, you can change the string or bit-image without having to change the code and recompile.

One example of a free string is the non-default state of a context-sensitive menu command. For example, GEM RCS's Global Menu has a command whose default state is **Hide Parts**. The default version of the menu is contained in the MENU tree; the non-default state of the command (**Show Parts**) is stored as a free string.

GEM RCS SCREEN

When you start GEM RCS, your screen looks like the illustration below. The labeled components are described after the illustration.

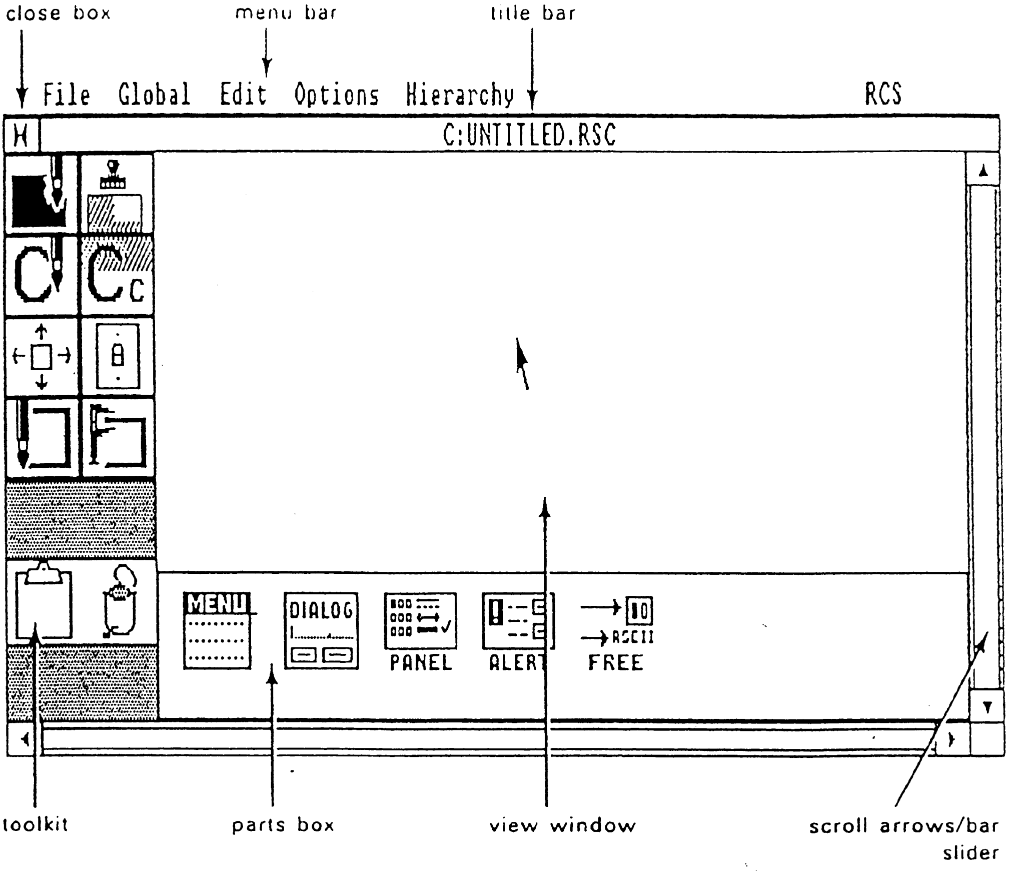


Figure 1-1. GEM RCS Screen

Menu Bar

The menu bar is the top line of your screen. It contains the menu titles File, Global, Edit, Options, Hierarchy, and RCS.

When you touch a menu title with the pointer, the menu drops down below the menu bar. You can then choose a command by clicking on its name in the menu.

GEM RCS's menu commands are described in detail in Section 3. Most of the commands can also be executed by typing one of the keystroke combinations listed in Table 3-4.

Title Bar

The title bar shows which of GEM RCS's two levels you're working in:

- root level
- object level

The root level is the highest level of the resource file. At this level, the title bar identifies the current directory path and the resource filename--**C:\TOOLS\MYAPP.RSC**, for example--and the view window contains the icons for the object trees (MENU, DIALOG, etc.) you have placed in the resource file.

The object level is the level at which you actually work on the objects in a tree--for example, adding text strings, command entries, or exit buttons. At this level, the title bar contains the tree's name (FILEMENU, for example), without any directory path information.

Note the following about the name of your resource file:

- Each resource file starts out as **UNTITLED.RSC** until you save it and give it a name.
- The resource file extension is .RSC (resource). Don't confuse it with the application's name: GEM RCS (Resource Construction Set).

Close Box

Clicking on the close box does the following:

- If you are at the object level, the close box closes the object and returns you to the root level of the resource file.
- If you are at the root level, the close box closes the resource file. If you have edited the file but not saved the edits, GEM RCS displays a dialog asking if you want to abandon the edits or save the file.

Note: If you are working on a new file that hasn't yet been named and saved, the close box has no effect at the root level.

The close box and the **Close** command on the File Menu can be used interchangeably.

Toolkit

The toolkit (described in detail in Section 3) contains icons for the various tools you can use to "customize" the objects in a tree. For example, you can change an object's color, fill pattern, or alignment with the tools.

View Window

The view window is where you design the layout of a tree and customize its objects.

Parts Box

The parts box (described in detail in Section 3) contains icons for the objects you can include in an object tree. The contents of the parts box change according to the type of tree you're working on.

Scroll Bars and Sliders

GEM RCS's scroll bars and sliders work in the same manner as the scroll bars and sliders on GEM Desktop windows. Note that in most cases you can scroll horizontally as well as vertically.

MOUSE TECHNIQUES

Many mouse techniques you use with GEM RCS, such as clicking and dragging, are the same as those you use with the GEM Desktop and other GEM applications. However, the effects of these techniques are often specific to GEM RCS, as the following table illustrates:

Table 1-1. GEM RCS Mouse Techniques

Technique	Effect
click	Selects tree or object. GEM RCS indicates a selected tree by highlighting its icon in reverse video and a selected object by defining its "extent" with a dotted line.
Shift-click	Selects more than one object.
Ctrl-click	Selects parent of object.
double-click	Opens tree or object for editing.
drag	Copies tree or object from parts box to view window; moves tree or object inside view window; moves tree or object to or from clipboard.
Ctrl-drag	Moves object's parent and all that parent's children.
Shift-drag	Copies tree or object inside view window; copies tree or object to or from clipboard.
Ctrl-Shift-drag	Copies object's parent and all that parent's children.

Note: When using any of the dragging techniques to move or copy trees or objects, remember the following:

- If you are copying from the parts box or if you are moving or copying within the view window, the dotted line extent of the tree or object must be entirely inside the view window or a parent object. Here are two specific examples:

- Dragging a tree icon from the parts box to the view window: if any part of the extent is in the title bar or the toolkit when you release the mouse button, the tree is not copied to the view window.
- Dragging a MENU tree ENTRY from the parts box to the menu box in the view window: if any part of the ENTRY extent is outside the menu box (its parent) when you release the mouse button, the ENTRY is not copied to the menu box.

- If you are moving or copying a tree or object to the clipboard, drag from the upper left corner of the tree or object. If you don't, and if any part of the extent is off the left edge of the toolkit, the tree or object is not moved or copied to the clipboard. (The clipboard is described in Section 3.)

Note also that when you copy a tree or object, the copy does not retain the tree or object names associated with the original.

Sizing Objects

To change the size of an object, first select the object. GEM RCS highlights the selected object with a dotted outline and displays the object's "size handle" (a solid black rectangle) in the bottom right corner. Then place the pointer on the size handle and drag the size handle until the object is the size you want.

If the object has children, note that GEM RCS won't permit you to make a parent object too small to contain its children.

Deleting Trees or Objects

To delete a tree or object, do any of the following:

- Drag it from the view window to the trash can icon in the toolkit. When the trash can icon is highlighted, release the mouse button.
- Click on the tree or object to select it. Then move the pointer to

the trash can icon (without dragging the tree or object), and click on the trash can.

- Select the tree or object and then choose the **Delete** command from the Edit Menu.

You cannot retrieve an item once you place it in the trash can.

WORKSPACE AND MEMORY USE

When-you start GEM RCS, it requests a portion of your computer's memory as its workspace. Depending on how much RAM is available, this workspace can be as much as 64K.

As you add trees and objects, you whittle away at the available workspace. However, if you delete a tree or object, GEM RCS does not restore the portion of the workspace occupied by the tree or object. The only way you can regain that memory is by saving the file. The **Save** and **Save As...** commands (described fully in Section 3) write out and reload the resource.

To find out how much of the workspace is available, choose the **Info...** command from the Options Menu. (Menu commands are described fully in Section 3.) The information dialog tells how many bytes the file, tree, or object has used and how much space remains in the workspace.

Note: The file size shown in the information dialog is only an approximation of the size of the resource file GEM RCS will produce, for the following reasons:

- Alerts actually take less space in the final file because they are written out as ASCII strings.

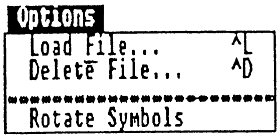
- Shift-dragged copies of objects might use more space in the final file.

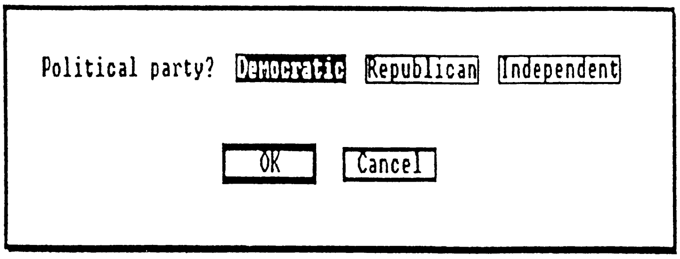
End of Section 1

Section 2

GEM RCS Tutorial

In this tutorial introduction to GEM RCS, you will start GEM RCS and then create the menu and dialog box shown below.





You will then save the resource file and exit GEM RCS.

STARTING GEM RCS

To start GEM RCS, start the GEM Desktop and then do the following:

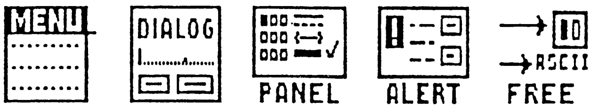
1. Open the TOOLS folder and locate the RCS.APP icon.
2. Double-click on the RCS.APP icon.

CREATING A MENU.

The menu you will create in the following steps contains simple text string commands. This part of the tutorial demonstrates opening a tree, adding objects to the tree, setting attributes, and sorting objects.

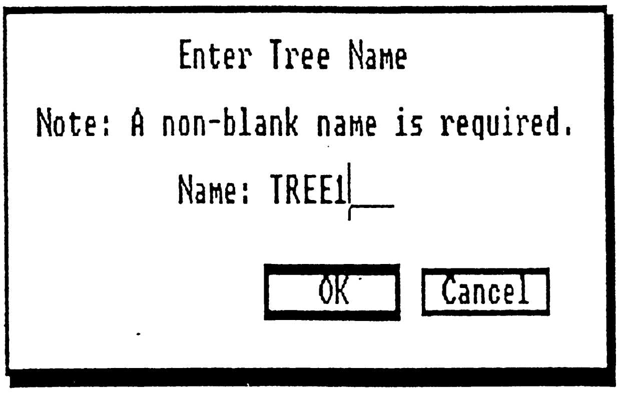
Choosing and Naming a MENU Object Tree Icon

The parts box contains icons for the five kinds of object trees you can construct with GEM RCS:



Drag the MENU icon into the view window. You don't have to locate the icon precisely; when you release the mouse button, GEM RCS automatically places a copy of the icon in the upper left corner of the window. (Remember that GEM RCS does not copy the icon to the view window if any part of the outline you're dragging is in the toolkit or the title bar.)

Whenever you place a new object tree in the view window, GEM RCS displays a dialog that asks you to name it.



Enter Tree Name

Note: A non-blank name is required.

Name: TREE1

Note that the dialog already contains the name TREE1. To use this name for your object tree, you would simply click on the dialog's OK button or press the Enter key. In this case, however, give the object tree a different name. Press the Esc key to erase TREE1, type the name **APPMENUS**, and then click on the OK button or press the Enter key. (See your GEM Desktop guide for a full description of entering and editing text in dialogs.)

Opening the Object Tree

Double-click on the APPMENUS icon. The following changes take place:

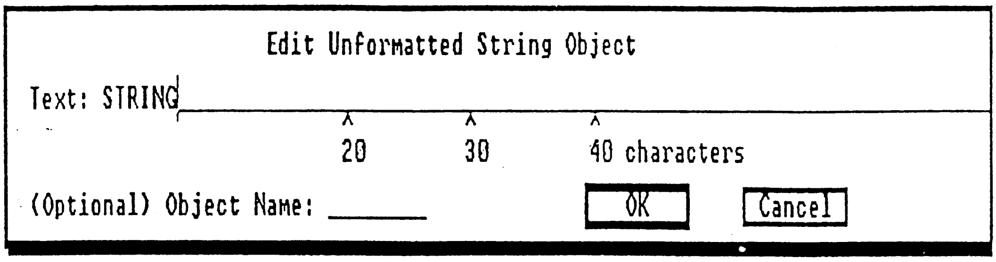
- The title bar now says APPMENUS.
- The view window contains a menu bar with two titles--File and Desk--and a shaded background area. (The Desk Menu is at the extreme right of the menu bar.)
- The parts box contains the four objects that can appear in the MENU object tree: the menu TITLE, an ENTRY (each command text string is an ENTRY), a line pattern to separate entries, and a box in which you can place menu items that are not text strings (fill patterns or line styles, for example).

In this tutorial you'll create a new menu from scratch. In Section 3 we'll tell you how you can use the existing File Menu and Desk Menu.

Adding a Menu Title

Drag the TITLE object from the parts box to the menu bar, placing it after "File". Be careful not to drag into the GEM RCS title bar or the toolkit. If you do, GEM RCS cancels what you've dragged.

Now double-click on TITLE. The following dialog appears on your screen:



Edit Unformatted String Object

Text: STRING

20 30 40 characters

(Optional) Object Name: ______

Do the following to edit and name the menu's title:

1. Press the Esc key to erase the text.

2. Press the spacebar, type **Options**, and then press the spacebar again. (If you don't press the spacebar, there won't be enough blank space between menu titles.)
3. Move the cursor to the Object Name field and type **OPTITLE**.
4. Click on the OK button or press the Enter key.

Adding Entries to the Menu

To add entries (commands) to the menu, do the following:

1. Click on the menu's title, Options. This selects the title and also displays a small menu box below the menu bar.
2. Place the tip of the pointer just inside the lower right corner of the menu box and press the mouse button. When the pointing finger icon appears, drag down and to the right to make the menu box larger. Make it larger than you think you need; you'll make it the right size later.
3. Drag an ENTRY from the parts box and place it in the upper left corner of the menu box. Don't crowd the corner too tightly, and make sure the outline you're dragging is entirely inside the menu box.
4. Copy this ENTRY by Shift-dragging. Place the copy just below the first ENTRY.
5. Drag a separator line from the parts box and place it just below the two entries.
6. Shift-drag the ENTRY once more, placing the new copy below the separator line.

Editing the Entries

The next step is to convert each generic ENTRY into a command an end-user can choose. To do so, do the following:

1. Double-click on the first ENTRY in the menu. GEM RCS displays the Edit Unformatted String Object dialog.

2. Press the Esc key to erase the text in the dialog.

3. Press the spacebar twice (to put two blank spaces before the command) and then type **Load File....** (The three dots are a GEM convention indicating that choosing the command will cause a dialog to be displayed.) Don't click on the OK button yet.

4. Many GEM applications use keystroke equivalents for their menu commands. Let's say the end-user could type Ctrl-L to produce the same effect as choosing the **Load File...** command. To indicate this in the menu, press the spacebar several times and then type ^L, followed by a single blank space.

5. Move the cursor to the Object Name field and type **OPSLOAD**.

6. Click on the OK button or press the Enter key.

Note to Step 4: GEM application menus follow the convention that ^ represents the Ctrl key and that a filled diamond represents the Alt key. (The keystroke combination **Ctrl-G** produces a filled diamond in the Edit Unformatted String Object dialog.) GEM application menus also follow the convention of surrounding the command text string with two leading blank spaces and a single trailing blank space.

Take the same steps for the second ENTRY, with the following variations:

- The command is **Delete File....**
- The keystroke equivalent is **^D**.
- The Object Name is **OPSDELT**.

Don't forget the leading and trailing spaces, and don't worry if the ^L and ^**D** don't line up properly at first. You can fix that simply by adding or removing spaces in the Edit Unformatted String Object dialog. The editing techniques are the same as for any GEM application dialog.

Now edit the third ENTRY, with the following variations:

- The command is **Rotate Symbols**.
- It has no keystroke equivalent.
- The Object Name is **OPSROTE**.

Sizing the Menu Box

To make the menu box the right size, first select the box by clicking inside it away from the commands or the separator line. You'll know you have been successful if the size handle appears at the box's lower right corner.

Next, drag the size handle until the menu box is the size you want.

You can also size the menu box without first selecting it. Use the same technique described in Step 2 under "Adding Entries to the Menu."

Sizing the Commands

Because GEM AES only highlights and enables the area contained within the extent of an ENTRY, you should make each command's extent match the full width of the menu box. If you don't, your application's menus will have a visual and functional inconsistency that the end-user will probably find displeasing.

To size a command, you can do any of the following:

- Select it and drag its size handle all the way to the right border of the menu box.
- Select it, display the alignment pop-up menu (see left), and choose the **Fill Horizontal** option.
- In the Text field of the Edit Unformatted String Object dialog, put blank spaces after the string value of each ENTRY.

Extending the Separator Line

To extend the separator line all the way across the menu box, do the following:

1. Double-click on the separator line. GEM RCS displays the Edit Unformatted String Object dialog. The heavy bars in the Text field of the dialog are equivalents to the shaded separator line in the menu.

2. To extend the line, type several Ctrl-S combinations in the Text field.

3. Click on the OK button or press the Enter key.

If the line is not long enough, repeat these steps until it is. If the line is now too long, GEM RCS displays an alert that this object no longer fits inside its parent. Click on the dialog's OK button and then double-click on the separator line again. Use the Backspace key to erase the bars until the separator line is the right length.

Sorting the Objects in the Menu

The last thing you should do before closing a tree is sort the objects in it. If you don't, at run-time your application will draw the objects at each level of the tree in the order in which you created them, which may be visually displeasing. By sorting the objects, you can determine the order in which they will be drawn.

To sort the objects in your menu tree, do the following:

1. Ctrl-click on any of the commands. This selects the menu box.

2. Display the Hierarchy Menu and choose the **Sort Children...** command. GEM RCS displays a dialog with four sorting options: single-row, single-column, double-column, double-row.

3. Choose the single-column option (second from the left) and press the Enter key or click on the OK button.

Closing the MENU Object Tree

Your menu is now complete. To close its object tree and return to the root level of the resource file, click on the close box or choose the **Close** command from the File Menu.

CREATING A DIALOG BOX

The dialog you will create in the following steps contains a text string, a set of "radio buttons," and two exit buttons. In addition to demonstrating more about setting attributes and sorting objects, this part of the tutorial illustrates working with a three-level tree, where one of the children of the root object has children of its own.

Choosing and Naming a DIALOG Object Tree Icon

The first step in creating a dialog box is the same one you took in creating the menu: choosing and naming the object tree icon.

Drag the DIALOG object icon from the parts box and place it in the view window. When GEM RCS displays the naming dialog, press the Esc key to erase the default name (note that it is TREE2; the MENU was TREE1), type **PTYDIAL**, and click on the OK button or press the Enter key.

Opening the Object Tree

To open your dialog box's object tree, double-click on the PTYDIAL icon. Note the following changes to your screen:

- The name of your object tree, PTYDIAL, appears in the title bar at the top of the view window.
- The view window is now empty.
- The icons in the parts box change, showing you the objects you can include in your dialog box.

Choosing Objects for Your Dialog Box

To assemble the objects for your dialog box, do the following:

1. Drag the STRING object from the parts box, placing it in the upper left of the view window.

▭

2. Drag the hollow box object (see left) from the parts box and place in the upper center of the view window.

3. Click on the box to select it. Note the object's size handle (mentioned under "Sizing Objects" in Section 1) at the lower right corner of the extent.

4. Drag the box's size handle down a little and almost to the right border of the view window, making a long, shallow rectangle.

5. Drag the BUTTON object from the parts box and place it inside the box you just enlarged, in the upper left corner.

Customizing the STRING Object

Double-click on the STRING object in the view window. When GEM RCS displays the Edit Unformatted String Object dialog, press the Esc key to erase the text and then type the following:

```
Political party?
```

You don't need to do anything with the optional object name, so click on the OK button or press the Enter key to complete your work in the Edit Unformatted String Object dialog. The STRING object in your dialog box now contains the phrase you just typed. (Don't be concerned if the message overlaps the box or BUTTON object. We'll reposition everything later.)

Customizing the BUTTON Object

To change the text inside the BUTTON object and to give it a name to which you can refer in your program code, do the following:

1. Double-click on the button. GEM RCS displays the Edit Unformatted String Object dialog.

2. Press the Esc key to clear the existing text.

3. Type **Democratic.**

4. Click on the "Object Name" field. The text cursor moves to this line. (You can also move the cursor by pressing the Tab key or the down-arrow key.)

5. Type **DEMBTN.** (To make reading your code as easy as possible, you should use object names that indicate the object's function.)

6. Click on the OK button or press the Enter key.

Setting a Radio Button

The "Democratic" button and two others (you'll create them in just a moment) will make up a set of "radio buttons." Like the pushbuttons on a car radio, radio buttons have the following characteristics:

- In a set, only one button at a time can be selected, but there should always be a selected button. There should never be a case where no button is selected.
- Selecting a button automatically de-selects the previously selected button in the set.

To set the button as a radio button, do the following:

1. Click on the "Democratic" button to select it.
2. Click on the attributes menu icon (see left) and click on "Radio Button" in the pop-up menu that appears.

Creating Additional Radio Buttons

This dialog has two additional radio buttons, "Republican" and "Independent". To create these buttons, do the following:

1. Shift-drag the "Democratic" button twice, placing the two copies to the right of the original.
2. Double-click on the copy immediately to the right of the original. In the Edit Unformatted String Object dialog, change the text to "Republican" and the object name to "REPBTN".
3. Double-click on the last copy. In the Edit Unformatted String Object dialog, change the text to "Independent" and the object name to "INDBTN".

Designating the Selected Radio Button

When the dialog appears at run-time, one of the radio buttons must be pre-selected (highlighted in reverse video). To pre-select the "Democratic" button, do the following:

1. Click on the button to select it.
2. Click on the attributes menu icon and choose "Selected".

Customizing the Parent Box

Before you work on the parent box, you might want to know why you need one in the first place.

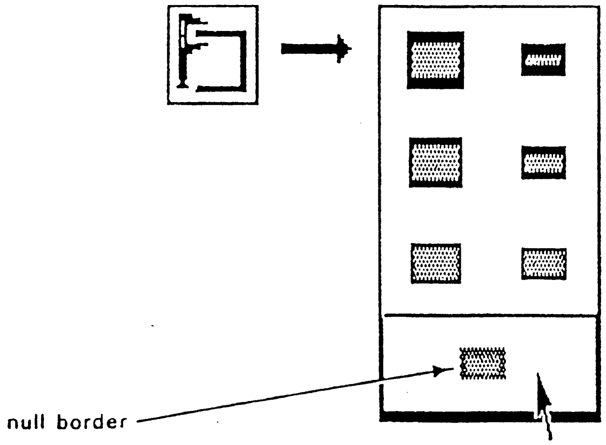
When you have more than one set of radio buttons in a dialog, you must enclose each set inside a parent box. If you don't, clicking on one button will turn off all other radio buttons in the dialog, not just the buttons in that one set.

In this dialog, with only a single set of radio buttons, you don't really need a parent box, but the dialog could have additional sets labeled "Age group?" and "Salary range?" This step will demonstrate what to do when you do need the parent box.

First, arrange the radio buttons in the parent box. (The "Democratic" button should be in the box's upper left corner.) Then select the parent box (if it isn't already selected) and size it down as small as you can make it.

You don't want the parent box to be visible in the final dialog. To make it invisible, do the following:

1. Select the box, if it isn't already selected.
2. Click on the border width menu icon (see below).
3. In the border width pop-up menu, choose the null border. See the illustration on the next page.



If you de-select the parent box by clicking elsewhere in the view window, you'll see that the parent is now invisible. However, you can still select it by Ctrl-clicking on one of the radio buttons, and you can move it (and the buttons too) by Ctrl-dragging any of the buttons.

Creating Exit Buttons

The dialog needs two "exit buttons" with which the end-user can exit the dialog by clicking on either button.

To create the exit buttons, do the following:

1. Drag a BUTTON object from the parts box, placing it below the radio buttons.
2. Shift-drag this BUTTON object, placing the copy just to the right of the original.
3. Edit and set attributes for the first BUTTON as follows:
   - Change its text to "OK".
   - Name it **OKBTN**.
   - Set its attributes to "Exit" and "Default".

   Making a button the "default" means that at run-time the end-user can select this button either by pressing the Enter key or by clicking on it. (You've already encountered a default button in the Edit Unformatted String Object dialog: the OK button.)

4. Now edit and set attributes for the second exit button as follows:
   - Change its text to "Cancel".
   - Name it **CANCLBTN**.
   - Set its attribute to "Exit".

5. Finally, locate the exit buttons where you want them.

Resizing the Background of Your Dialog Box

Move the pointer to an open area of the view window (away from the text and buttons) and click. The dialog box's size handle appears in the bottom right corner of the view window.

Place the pointer on the size handle and drag up and to the left. As you drag, the background of your dialog box gets smaller. When the background is the size you want, release the mouse button.

Sorting Objects in the Dialog

The last thing to do before closing this tree is to sort the objects in it. The procedure is essentially the same as with the SAMPLE dialog.

Note: You must sort all levels (sets) of children. For example, this dialog has two levels. The radio buttons are children of the empty box, and the empty box, string, and exit buttons are children of the dialog's outer box.

To sort the children in your dialog, do the following:

1. Ctrl-click on one of the radio buttons. This selects its parent box.

2. Display the Hierarchy Menu and choose the **Sort Children...** command. Click on the single-row option (on the far left) and press the Enter key or click on the OK button.

3. Select the dialog's outer box.

4. Display the Hierarchy Menu and again choose the **Sort Children...** command. This time click on the double-row option (on the far right) and press the Enter key or click on the OK button.

Sorting the objects will draw them in the following order:

1. "Political party?" string.
2. Radio buttons, "Democratic" first.
3. OK button.
4. Cancel button.

Closing the DIALOG Object Tree

Your dialog is now complete. To close its object tree and return to the root level of the resource file, click on the close box or choose the **Close** command from the File Menu.

SAVING A RESOURCE FILE

Display the File Menu and choose the **Save As...** command. The Item Selector appears on your screen.

Type **TEST** and then click on the OK button or press the Enter key. GEM RCS saves the following files:

TEST.RSC The actual resource file you would include with your application program. Note that you do not need to type the .RSC file extension; GEM RCS automatically supplies it.

TEST.DFN An auxiliary file that identifies the trees and objects in your resource file. When you work in GEM RCS, you need both the resource file and its .DFN file in the same directory.

If you have created.resource files with GEM RCS, the earlier version of the Resource Construction Set, the definition file has the extension .DEF. To use this file with GEM RCS, simply change its extension to .DFN.

In its initial default configuration, GEM RCS also creates TEST.H, a C language include file you must use when you compile your application program code. The .H file is an ASCII file that lists the trees and objects in your resource file and the object numbers assigned to them by GEM RCS. You can also read this file to check the order in which objects have been sorted. Sorting is discussed in this tutorial and also in the description of the Hierarchy Menu in Section 3.

Note: You can also produce include files for Pascal, BASIC, and FORTRAN-77, as well as an editable ASCII version of your resource file (the .RSH file). See the description of the **Output...** and **Save Preferences** commands in the Global Menu (Section 3) and the description of RSCREATE in Section 4.

QUITTING GEM RCS

To stop GEM RCS, choose the **Quit** command from the File Menu. You then return to the GEM Desktop.

End of Section 2

Section 3

GEM RCS Reference

ASPECT RATIO AND SCREEN RESOLUTION

GEM applications can run under a variety of aspect ratios and screen resolutions. The following table lists the three most common combinations.

Table 3-1. Screen Resolution and Aspect Ratio

Resolution	Pixel Ratio	Aspect Ratio
low	640x200	3.2:1
high*	640x400	1.6:1
high*	720x350	2.06:1
square*	any	1:1

*Although their aspect ratios are different, the GEM software uses the same screen driver for these resolutions.

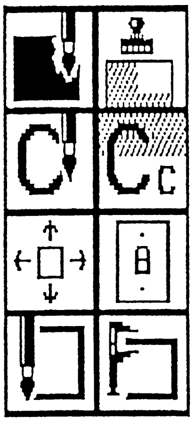
If an icon or bit image is displayed on a system with an aspect ratio different from the system on which it was created, the icon or bit image will appear taller or shorter than it appeared in the original. This can cause problems if you are placing icons or bit images close together; aspect ratio differences can cause them to overlap each other or separate from each other.

You can resolve this issue in either of the following ways:

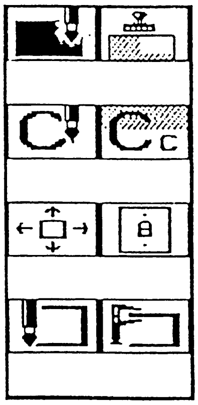
- You can create a single resource file and compensate for the different aspect ratios of the systems on which you expect your application to run.

- You can create different sets of icons and bit images for each expected aspect ratio. You can then create aspect ratio-specific resource files from a single "master" file by loading the appropriate set of icons and bit images into each final version of the resource file.

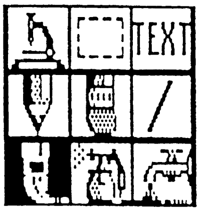
The following illustrations demonstrate these two approaches. On the left, you see a partial view of the GEM RCS toolkit, which was created only in a low-resolution version. Note how the bit images are flattened on a high-resolution system. On the right, you see partial views of the GEM Paint^{TM} toolkit, which exists in both high- and low-resolution versions.



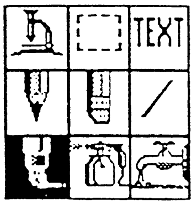
GEM RCS
lo-res



GEM RCS
hi-res



GEM Paint
lo-res



GEM Paint
hi-res

If you create aspect ratio-specific resource files, at run-time your application can determine the screen's aspect ratio and resolution and load the appropriate file.

For more on icons and bit images, see "Icons and Bit Images," later in this section.

Note: Aspect ratio differences do not cause problems with text or any other objects in a tree.

FILE AND DESK MENUS

As noted in the tutorial, when you open a MENU object tree, there are already a File Menu and a Desk Menu in the menu bar.

File Menu

The File Menu is optional; you can drag its title to the trash can.

To add commands to the File Menu, do the following:

1. Click on the menu title. This selects the title and displays the menu box with a single command, **Quit**.
2. Place the pointer at the lower right corner of the menu box, and Ctrl-drag down and to the right. This makes the menu box larger, without affecting the size or location of the command.
3. Drag the **Quit** command to the bottom of the menu box, or place it wherever you want it.

From this point, you can create the menu just as you did in the tutorial.

Desk Menu

This menu is required; you cannot drag its title to the trash. However, at run-time its name and location can vary as follows:

- If your application is running under version 1.X of GEM AES, the menu is called the Desk Menu and appears at the left side of the menu bar.
- If your application is running under version 2.X of GEM AES, the menu appears at the right side of the menu bar, and the menu's name is the filename of the application's executable file. For example, for an application executed by a file called BINGO.APP, it would be the BINGO Menu.

The name variation and menu placement are handled entirely by the GEM AES.

The Desk Menu contains several "placeholder" commands. The first--

Your message here--is fully editable (the message has a 20-character maximum), but you cannot delete it. A typical function for this command is to display an informational dialog about the application, including its version number.

The numbers below the separator line are placeholders for desk accessories. The menu can accomodate a maximum of six. You cannot edit or delete these entries. GEM AES handles placing desk accessory names in the menu and sizing the menu according to the number of desk accessories it contains.

USING THE PARTS BOX

The parts box contains icons for the objects you can include in an object tree. These objects have preassigned attributes (fill patterns, colors, text strings, etc.) you can change with the tools in the toolkit. You can edit text in an object or make other changes by double-clicking on the object to display one of a variety of dialogs.

To place an object in your current object tree, drag its icon from the parts box to the view window.

The contents of the parts box depend on the type of tree you are creating, as the following sections explain.

MENU Tree Objects

TITLE Menu title for the menu bar.

ENTRY Command string for the menu.

▒▒▒▒▒▒▒▒ Separator line for the menu. Use the separator line to separate commands into logical groups for the end-user's convenience.

□ Hollow box you can place in a drop-down menu. You can use the hollow box to define the selectable (reversible) extent of a bit image in the menu. For an example, see the Gallery Menu in the GEM Graph^{TM} application.

DIALOG Tree and PANEL Tree Objects

The same set of objects is available for DIALOG and PANEL trees. In the following descriptions, the names in parentheses are object types described in Section 6, "Object Library," in the GEM AES Reference Guide.

BUTTON (G_BUTTON) - Boxed string the end-user selects to indicate a choice among alternatives.

STRING (G_STRING) - Boxless character string, normally containing explanatory text for the end-user.

EDIT:_____ (G_FTEXT) - A formatted text field the end-user can change.

EDIT:_____ (G_FBOXTEXT) - A formatted text field inside a box. The end-user can change the text field.

□ (G_IBOX) - Hollow box through which the end-user can see the fill pattern or text beneath.

□ (G_BOX) - Opaque (non-transparent) box.

TEXT (G_TEXT) - Formatted text. You can select sizes, colors, fonts, and masks for these objects.

C (G_BOXCHAR) - Single character in an opaque box.

BOXTEXT (G_BOXTEXT) - Same as TEXT, but with a surrounding, opaque box.

ICON

(G_ICON) - Use to display an icon (data plus mask) created with GEM IconEdit.

IMAGE

(G_IMAGE) - Use to display a bit image (data field only) created with GEM IconEdit.

ALERT Tree Objects

Alerts use an optional warning icon, up to five message lines, and up to three exit buttons. The descriptions of the icons (below) also suggest how each kind of alert might be used. This "philosophy" is entirely optional.

Button

The default state of an exit button in an ALERT is set in your program code.

Message Line

An unformatted text string telling the end-user the nature of the problem. An alert can have a maximum of five message lines, each with no more than forty characters.

NOTE icon. This is the default icon. This kind of alert can inform the end-user that he has attempted something the application does not permit. The alert might only have an OK exit button, so the user can acknowledge the message.

?

WAIT icon. This kind of alert can inform the end-user that he has requested a permitted action, but that the action is not possible under the present circumstances, perhaps because a drive door is open or there is no disk in the drive. The message text can suggest solutions. The alert would have two exit buttons, Cancel and Retry.

STOP icon. This kind of alert can warn the end-user that his request, while perfectly legal, can result in the loss of data. ·Formatting a disk is one example. The alert would have two exit buttons, OK and Cancel:

FREE Tree Objects

Free-string

An unformatted text string containing the text you wish to display.

A bit image (data field only) created with GEM IconEdit.

OBJECT CLASSES

The objects in the parts box can be divided into five classes:

- unformatted strings
- box type
- formatted text
- bit images
- icons

The following table lists the object classes, the object types in each class, and the object trees in which they can appear.

Table 3-2. **Object Classes**

Object Class	Object Type	Object Tree(s)*
Unformatted String	TITLE	M
	ENTRY	M
	separator line	M
	BUTTON	D,P,A
	STRING	D,P
	Message Line	A
	Free-string	F
Box Type	all boxes, including outer box of DIALOG or PANEL	M,D,P
	single character in box	D,P
Formatted Text	TEXT	D,P
	BOXTEXT	D,P
	editable text (EDIT:____)	D,P
	boxed editable text	D,P
Bit Images	IMAGE	D,P,F
Icons	ICON	D,P

*M = MENU, D = DIALOG, P = PANEL, A = ALERT, F = FREE

Unformatted Strings and Formatted Text

GEM RCS provides two ways of entering text in object trees: unformatted strings and formatted text. The following table lists some of the characteristics of and differences between these two kinds of text.

Table 3-3. Unformatted Strings and Formatted Text

Unformatted Strings	Attributes can be set.
	Can be aligned within parent.
	System font and color only.
	Transparent mode only (background color and pattern show through).
	Always left-aligned in extent.
Formatted Text	Attributes can be set.
	Can be aligned within parent.
	Colored text available.
	Small font available.
	Transparent mode or replace mode (background color and pattern do not show through).
	Text can be left-aligned, right-aligned, or centered within extent.
	Outline color available (boxed formatted text only).
	Outline thickness available (boxed formatted text only).

ADDING OBJECTS TO TREES

To add an object to your tree, drag its icon from the parts box to the view window. You can then Shift-drag the object in the view window to make additional copies.

The Shift-drag technique is especially useful if you want two or more objects with the same attributes. For example, if you want three buttons with attributes in common, drag one BUTTON object from the parts box, set the common attributes, and then copy it twice, rather than dragging the object from the parts box three times and setting the attributes three times.

Special Cases

MENU trees

The TITLE object can be placed only in the title bar. The ENTRY, separator line, and hollow box objects can only be placed in the menu box.

You cannot add or remove objects from the Desk Menu. The first line of the Desk Menu is the only editable entry.

ALERT trees

An ALERT tree can only contain one warning icon. To replace the default NOTE icon, drag another icon from the parts box and place it inside the alert. GEM RCS automatically deletes the NOTE icon and positions the new icon. If you don't want an icon, drag the existing one to the trash can.

An ALERT can have a maximum of three exit buttons and five message lines of no more than forty characters each. GEM RCS automatically adjusts the size of the alert box to the length of the message text.

MOVING, COPYING, AND DELETING OBJECTS

Moving and copying objects are described in Table 1-1. Also see "Using the Clipboard," later in this section.

To delete an object, do one of the following:

- Drag it to the trash can.
- Select the object and click on the trash can icon.
- Select the object and choose the **Delete** command from the Edit Menu.

SIZING OBJECTS

You can size most objects by selecting the object and then dragging its size handle.

Some objects--including STRING objects and others whose contents you edit in the Edit Unformatted String Object dialog--change size automatically when you add to the text in the object. However, such a change might violate the parent-child size relationship, which is described next.

MAINTAINING THE PARENT-CHILD SIZE RELATIONSHIP

As detailed in Section 6 of the GEM AES Reference Guide, object structure rules require that the parent object always contain its children. This means that the child cannot be larger than its parent and that the child cannot extend past the boundaries of the parent.

In several cases, GEM RCS protects against violations of the parent-child size relationship. For example, you cannot size down a menu box so that the entries or separator lines extend past the edge of the box. Similarly, you cannot enlarge a BUTTON object to extend past the edge of a dialog box.

However, by editing text strings in the Edit Unformatted String Object dialog, you can make the following objects too long to fit inside their parent objects:

- a STRING in a dialog or panel
- an ENTRY object in a menu box

For example, you can type text into an ENTRY so that the command extends past the edge of the menu box. In this case, GEM RCS displays an alert warning you that the child does not fit inside its parent. You have two options:

OK GEM RCS accepts the long ENTRY and automatically enlarges the text string's extent to include the entire string. You should then make the menu box larger. (If you don't, be forewarned that GEM AES only writes to the menu/alert buffer and redraws the part of the screen under the menu box. Anything outside the menu box will remain on the screen as garbage.)

Cancel GEM RCS ignores your changes and retains the original ENTRY text.

ICONS AND BIT IMAGES

Icons and bit images are both created with GEM IconEdit.

An icon consists of DATA (the "picture") and MASK, usually a solid "shadow" of the DATA. The function of the MASK is to prevent any background colors or patterns from showing through the DATA. You can edit an icon to include a text string or a single character at any of several positions relative to the icon. See the description of the Edit Icon Object Dialog, later in this section.

A bit image is DATA only. Because it has no MASK, the bit image allows any background color or pattern to show through. Unlike icons, bit images cannot be edited to include text strings or characters.

Loading Icons

To load an icon into a DIALOG or PANEL tree, drag the ICON icon from the parts box to the view window. Then select the icon and choose the **Load...** command from the Options Menu.

GEM RCS first displays a dialog asking if you want to load the icon's DATA, MASK, or both. If you load DATA only, any background color or pattern will show through the icon. If you load MASK only, you will simply block out any background color or pattern, but the icon's image will not appear.

If you choose both DATA and MASK, GEM RCS displays the Item Selector twice. The first time, select the icon's DATA. When GEM RCS immediately redisplays the Item Selector, select the icon's MASK.

Note: The files for DATA and MASK both have the extension .ICN. You'll need some naming convention to distinguish the two, like ICOND.ICN and ICONM.ICN.

Loading Bit Images

To load a bit image into a DIALOG or PANEL tree, drag the IMAGE icon from the parts box to the view window. Then select the icon and choose the **Load...** command from the Options Menu. GEM RCS displays the Item Selector, and you can select the bit image file.

Like icons, bit image files have the .ICN extension.

USING FREE TREES

As noted in Section 1, FREE trees can be used to enable context-sensitivity in your application at run-time. We'll use GEM RCS as an example of how you can use Free-string objects and FREE images.

Note: You can assemble your Free-strings and FREE images in as many FREE trees as you like, mixing and matching as the spirit moves you. However, when GEM RCS saves the resource file, it combines all Free-strings into one tree (called FRSTR1) and all FREE images into another tree (called FRIMG1).

Free-String Objects

The GEM RCS resource file contains several strings in the tree FRSTR1, including the following commands:

- **Hide Parts**
- **Show Parts**

The default form of this command (**Hide Parts**) is contained in the tree for GEM RCS's menus. When the end-user chooses the command, the application changes the pointer for the command string from the default in the MENU tree to the alternate (toggled) command in the FRSTR1 tree. The next time the GEM AES screen manager draws the menu, it uses **Show Parts** from FRSTR1. From then on, as context requires, the application swaps in and out the pointers to the two forms of the command in FRSTR1.

Note: If you change the length or spacing of the string in the tree containing the default, don't forget to make the same changes to the related free-string or strings.

FREE Image Objects

The GEM RCS clipboard (described later in this section) is a context-sensitive object. Its default appearance is a bit image object in the toolkit's object tree. The tree FRIMG1 contains its alternate appearance (dogeared, to indicate that the end-user has placed something on the clipboard) and a duplicate of the default version.

When the end-user places something on the clipboard, GEM RCS swaps in the dogeared bit image from FRIMG1. The next time the default version is needed, GEM RCS uses the duplicate from FRIMG1, and from then on GEM RCS swaps in and out the two versions from FRIMG1.

USING THE OBJECT EDITING DIALOGS

When you open an object in the view window, GEM RCS displays one of the following dialogs:

- Edit Unformatted String Object
- Edit Box Type Object
- Edit Formatted Text Object
- Edit Bit Image Object
- Edit Icon Object

The dialog displayed depends on the class of object opened (see Table 3-2).

Each object editing dialog is different (see the individual descriptions following), but in each case you can use the dialog to name or rename the object you're working on. To do so, click on the line following "Object Name" and type the name you want to assign to the object. You can then refer to this name in your program code.

Edit Unformatted String Object Dialog

The default text, BUTTON or STRING, appears in the "Text" field. To edit the field, press the Esc key to erase the text in the field and then type the new text. You can move the cursor back and forth in the field with the left- and right-arrow keys, and you can erase characters with the Backspace and Delete keys.

To help you gauge the length of your text strings, the dialog marks the "Text" field at 20, 30, and 40 characters.

See "Unformatted Strings and Formatted Text," earlier in this section, for a discussion of the differences between unformatted and formatted text.

Edit Box Type Object Dialog

If the object is the single character in a box (object type G_BOXCHAR), the "Character" field contains the default character. You can edit or erase this character as you wish. For all other box objects, the "Character" field is empty, and GEM RCS ignores any characters you enter in it.

Edit Formatted Text Object Dialog

The Edit Formatted Text Object dialog exists in two forms: one for editable text (G_FTEXT and G_FBOXTEXT objects) and another for non-editable text (G_TEXT and G_BOXTEXT objects). The ob_spec value of all four object types is a POINTER to a TEDINFO structure (see Section 6 of the GEM AES Reference Guide).

See "Unformatted Strings and Formatted Text," earlier in this section, for a discussion of the differences between unformatted and formatted text.

The dialog for non-editable formatted text has only one field, PTEXT, in which you enter text.

The dialog for editable formatted text has three fields: PTMPLT (template), PVALID (validation), and PTEXT (text entry). These fields are explained at length in Section 6 of the GEM AES Reference Guide under "TEDINFO Structure." Note that in GEM RCS, PTMPLT, PVALID, and PTEXT use a tilde (~) in place of an underline (_) to avoid confusion between the field itself and the placeholders for editable characters.

The following example illustrates how PTMPLT, PVALID, and PTEXT create an editable text field in which the end-user can type the date.

```
PTMPLT>Today's date:  ~~/~~/~~____________________
PVALID>~~~~~~~~~~~~~~~~99~99~99____________________
 PTEXT>~~~~~~~~~~~~~~~~01~01~86____________________
```

PTMPLT: To edit the field, press the Esc key and then type the new string as shown above. The tildes take the place of the characters the end-user-can edit at run-time.

PVALID: The validation field controls the location and type of the characters the end-user enters at run-time. A "9" in PVALID indicates that the end-user may only enter a digit (0-9) in that position. Tildes in PVALID represent literal characters that must appear exactly as entered in PTMPLT.

PTEXT: The digits under the nines in the validation field appear as the default entry for the field. The end-user can type over them to change the date.

Edit Bit Image Object Dialog

This dialog contains only the "Object Name" field and a reminder to use the **Load...** command (Options Menu) to load the new bit image data. "Loading Bit Images," earlier in this section, describes how you load bit images.

Edit Icon Object Dialog

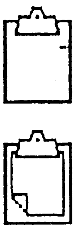
In addition to naming the object, you can use this dialog for the following:

- To enter and locate a text string you want to appear as part of the icon.
- To enter and locate a single character you want to appear as part of the icon.

The dialog contains two locator boxes: one for the Text field and another for the Character field. You can locate a text string at the top, middle, or bottom of the icon. You can locate a character at any of nine positions relative to the icon.

See "Loading Icons," earlier in this section, for a description of how you load an icon.

USING THE CLIPBOARD



The clipboard is a storage place for trees or objects from the view window. You can move or copy items to the clipboard, but only one item can be on the clipboard at a time. Moving or copying overwrites anything currently on the clipboard.

When you place an item on the clipboard, GEM RCS dog-ears the icon to let you know there is something there.

To cut a tree or object to the clipboard, do one of the following:

- Drag its icon to the clipboard.
- Select the tree or object and then click on the clipboard icon.
- Select the tree or object and then choose the **Cut** command from the Edit Menu.

Cutting removes the tree or object from the view window.

To copy a tree or object to the clipboard, Shift-drag its icon, or select the tree or object and use the **Copy** command on the Edit Menu. Copying leaves the original icon in the view window.

To paste a tree or object from the clipboard to the view window, do either of the following:

- Drag from the clipboard icon.
- Display the Edit Menu, press the mouse button when the **Paste** command is highlighted, and drag from the menu. Use this technique if you have removed the toolkit from the screen with the **Hide Tools** command.

Either technique empties the clipboard.

To copy a tree or object back to the view window, Shift-drag the clipboard icon. The clipboard still contains the original tree or object.

Whenever you paste or copy a tree from the clipboard, GEM RCS displays the "Enter Tree Name" dialog so you can give it a unique name.

You can use the clipboard as a holding place for an object from the parts box of one tree type and then add the object to another tree. For example, you can add bit images to a MENU tree by doing the following:

1. Open a DIALOG tree.
2. Put a bit image in the DIALOG tree's view window and then cut or copy it to the clipboard.
3. Close the DIALOG tree and open a MENU tree.
4. Paste or copy the bit image from the clipboard into a menu. You might want to make the bit image the child of a hollow box. See the description of the hollow box under "MENU Tree Objects," earlier in this section.

Note: Be careful when you mix and match objects this way. Some objects and trees are incompatible, and mixing them can cause unpredictable results. See "Unknown Object Tree Types" in Section 4.

GEM RCS clears the clipboard when you choose the **New** command to begin a new resource file and when you choose the **Open...** command to edit an existing resource file. Both commands are on the File Menu.

USING THE TOOLKIT

The toolkit contains tools with which you can make a variety of changes to a selected object, include the following:

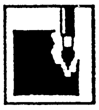
- changing the object's color
- changing the object's fill pattern
- aligning the object within its parent
- setting certain attributes for the object
- changing the object's line thickness

A tool is disabled if its effect is not meaningful for the selected object. For example, the fill pattern tool is disabled for STRING objects because STRING objects cannot have a fill pattern.

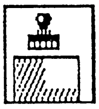
When you move the pointer over the toolkit, GEM RCS highlights only the tools enabled for the currently selected object. To use a tool, click on the highlighted icon. GEM RCS displays a pop-up menu of the tool's options.

To choose from a pop-up menu, drag through the menu and click when the option or command you want is highlighted. If you click outside the menu, it disappears, and the object is unaffected.

Descriptions of the Tools

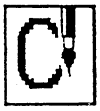


Selects a background color for an object. The menu appears in color if your computer can display color. Otherwise, assign colors using the numbered color codes. These correspond to the codes described under "Object Colors" in Section 6 of the GEM AES Reference Guide.

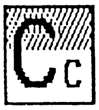


Selects a fill pattern for an object.

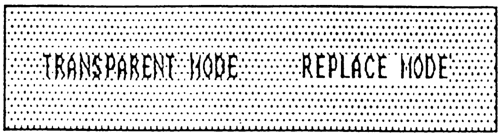
Note: For a fill pattern to be visible, it must also have a visible color. The defaults are white color and transparent fill pattern.



Selects a color for a TEXT, BOXTEXT, FTEXT, or FBOXTEXT object. The menu appears in color if your computer can display color.

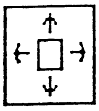


Aligns text, changes text size, or changes the background. For TEXT, BOXTEXT, FTEXT, and FBOXTEXT objects only. Most of the commands are self-explanatory; explanations and an illustration of **Transparent** and **Replace** follow.



Transparent – Makes each character cell (the text's immediate background) transparent; the color and fill pattern of the box containing the text shows through.

Replace – Makes each character cell opaque; the text appears on a solid white background.

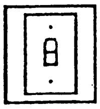


Aligns or resizes an object within the boundaries of its parent. The alignment options are self-explanatory; explanations of the fill and snap options follow.

Fill Horizontal – Automatically resizes the object so that it extends from the extreme left to the extreme right boundaries of the parent object.

Fill Vertical – Automatically resizes the object so that it extends from the top to the bottom boundaries of the parent object.

Character Snap – Snaps the selected object in a PANEL tree to the nearest character boundary. This is useful for precise alignment of objects in the PANEL, which otherwise allows free placement of objects.



Selects attributes for an object. GEM RCS places a marker next to any attribute assigned to the current object. The attributes in the upper half of the menu set object flags (ob_flag) and affect the object's interaction with the FORM_DO call. The attributes in the lower half of the menu set object states (ob_state) and affect how the Object Library draws the object. See Section 6, "Object Library," and Section 7, "Form Library," in the GEM AES Reference Guide.

Explanations of the attributes follow.

Selectable - The end-user can select the object by clicking on it. Clicking displays the object in reverse video.

Exit - Clicking on the object fulfills an exit condition and causes FORM_DO to finish processing and return a value. Typically assigned to the exit buttons in a dialog. Objects with the Exit attribute must also have the Selectable attribute. They can also have the Default attribute, but that is optional.

Default - The object is selected automatically when the end-user presses the Enter key at run-time. If you designate more than one object in a tree as the default, you won't receive an error, but your results are not predictable.

Note: The Exit and Default attributes each add a pixel to all sides of the border of a BUTTON object. Thus, a button with only the Exit attribute has a two-pixel border, and a button with both attributes has a three-pixel border.

Radio Button - Makes the object a member of a set of "radio buttons." Radio buttons are like the buttons on a car radio; pressing one button makes another one pop out, which means the end-user can only select one radio button at a time. Members of a set of radio buttons must be nested at the same level within a common parent object.

Touchexit - Pressing the mouse button while the pointer is on the object fulfills an exit condition and causes FORM_DO to finish processing and return a value. The application does not wait for the mouse button to come up. Objects with the Touchexit attribute must not have the Selectable attribute.

Editable - The end-user can change information in the object at run-time. Use only with editable text objects.

Hidden - Hides the object so that it is not displayed on the screen. Use the **Unhide Children** command on the Hierarchy Menu to display hidden objects.

Shadowed - Draws a drop shadow around the object (usually a box). Shadows and outlines (see below) conflict.

Checked - Draws a triangle inside the left margin of the object.

Outline - Draws an outline around a boxed object. Outlines conflict with shadows and have no effect on.boxes with outward borders.

Crossed - Draws an X through the object in the system background color. Usė only with boxed objects.

Disabled - Draws the object at half-intensity (gray). The end-user cannot select a disabled object.

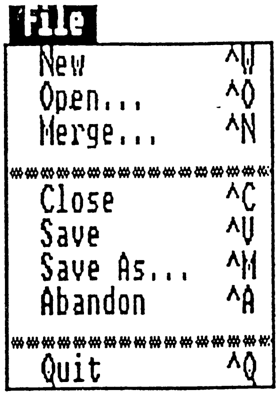
Selected - The object is preselected (appears in reverse video) when the end-user sees the tree.

Selects a color for the border around a boxed object. The menu appears in color if your computer can display color.

Selects a border thickness for any boxed object but a BUTTON. The default thickness is a single pixel around the outside of the object (the bottom of the left column in the menu). The left column also provides two- and three-pixel thicknesses outside the object. The right column provides one-, two, and three-pixel thicknesses inside the object. The option in the lower center of the menu is the null (invisible) border. The fill patterns in the pop-up menu are for illustration only and do not affect the fill pattern in the object. See the illustration in Section 2.

GEM RCS MENU COMMANDS

Note: Commands are disabled (dimmed) when choosing them is not meaningful in the present context of GEM RCS. For example, the Edit Menu commands (**Cut**, **Copy**, **Paste**, and **Delete**) are disabled when no tree or object is selected.



New Clears the view window so you can start a new resource file.

Open... At the root level only, and with no tree icon selected, displays the Item Selector so you can open an existing resource file.

If you have a tree or object currently selected in the view window, this command opens that tree or object.

Merge... Displays the Item Selector so you can select an existing resource file to merge with the one on which you are working.

Merging resource files can produce name conflicts. If the merged file contains names already used in the current file, GEM RCS creates new names for the duplicates. (Printing the .H or .I file is a good way to check for duplicates--see the **Output...** command in the Global Menu.)

Close Closes the current resource file or tree. Closing a file clears it from the screen. (If you have not saved your current edits, GEM RCS displays a dialog asking whether to save or abandon them. See the note below.) Closing a tree returns you to the root level of the file. Clicking on the close box has the same effect as choosing this command.

Save Saves your current resource file. The file remains in your workspace so you can continue editing. Use this command periodically so you won't lose all of your work to a power failure or computer malfunction.

Save As... Saves your current resource file under a name you provide. Use this command in either of the following situations:

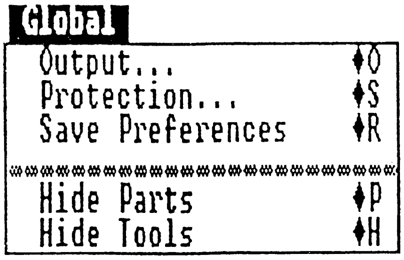
- To name and save a resource file for the first time.
- To save an edited version of an existing file under a new name and/or to a different directory. The new directory path and/or filename appear in the title bar. The original version of the file remains on disk under the old name.

Abandon Abandons all edits on the current file since the last time it was saved.

Quit Stops GEM RCS and returns you to the GEM Desktop.

Note: If you edit a new or existing resource file and then choose the **New**, **Open...**, **Close**, **Abandon**, or **Quit** command without first saving your edits, GEM RCS displays a dialog that asks if you want to abandon your edits. The dialog's three exit buttons offer you these options:

- Abandon the edits.
- Save the file.
- Cancel the requested command.



Output...

Selects the types of output files (in addition to the .RSC and .DFN files) GEM RCS creates when you save resource files. You can select include files for C (the default), Pascal, BASIC, and FORTRAN-77, as well as a .RSH source file for use with RSCREATE (see Section 4).

Protection...

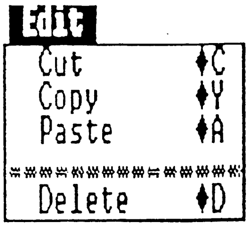
Sets the level of protection GEM RCS applies while you are editing resource files. This command has the following options:

- LOCKED - Permits editing and sizing but forbids changes to the tree structure. This setting is intended for post-production changes, such as translating strings for internationalization of your program. Values associated with tree and object names are preserved; you need not recompile the application after editing.
- NORMAL - The default protection level. It warns you before rearranging trees. NORMAL permits all operations but warns you of impending changes in parent-child relationships and workspace clearance.
- EXPERT - Gives you no warnings. You can make whatever changes you wish to the resource file.

Save Preferences Makes the settings and selections in the output and protection dialogs the defaults for subsequent GEM RCS sessions; saves them to a file called RCS.INF in the GEMAPPS folder. To restore the original defaults, delete RCS.INF.

Hide Parts Hides the parts box on your screen. The command toggles between **Hide Parts** and **Show Parts**.

Hide Tools Hides the toolkit on your screen. The command toggles between **Hide Tools** and **Show Tools**.

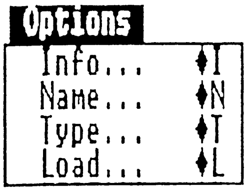


Cut Cuts the currently selected tree or object to the clipboard. The original no longer appears in view window.

Copy Copies the currently selected tree or object to the clipboard. The original remains in view window.

Paste Lets you move a tree or object from the clipboard to the view window by dragging from the Edit Menu. (Use when toolkit is not visible.)

Delete Deletes the selected tree, object, or group of objects.

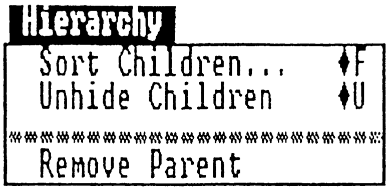


Info... Shows you information about the current file or the currently selected tree or object. Also shows how much of your workspace you have used and how much you still have available.

Name... Displays a dialog you can use to name or rename an object tree.

Type... Displays a dialog you can use to change a tree or object's type.

Load... Displays the Item Selector so you can replace an IMAGE or ICON object with one created using GEM IconEdit.

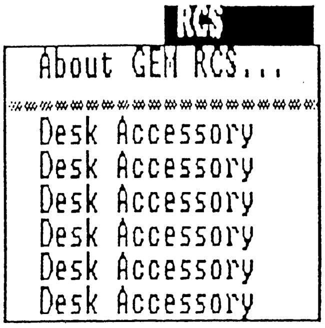


Sort Children... Displays a dialog in which you can specify how objects should be sorted within their parent. Sorting causes the objects in the tree to be drawn in a logical order. Without sorting, the objects in the tree are drawn in the order in which they were added to it. This has no effect on the functionality of the tree, but it can be visually displeasing,

Unhide Children Causes any hidden children of the currently selected object to be displayed.

Remove Parent Removes the currently selected object without removing its children. (Dragging an object to the trash also removes its children.)

Note: Resorting children after your application has been compiled can renumber the objects in your tree, causing any references to them in the program code to be incorrect. (You can check this by looking at the .H file created by GEM RCS.) If incorrect references exist, the program code must be recompiled.



About GEM RCS... Displays a dialog showing information about GEM RCS, including the version number, copyright, and authors' names.

Desk accessories Names of desk accessories currently available on your system. The menu can display up to six desk accessory names.

KEYSTROKE EQUIVALENTS OF MENU COMMANDS

Most of the GEM RCS commands can be executed by clicking on the command in the menu or by typing a keystroke combination. The following table lists the commands in the order they appear in the menus and gives their keystroke equivalents.

Note: A command's keystroke equivalent is only enabled if the command itself is enabled.

Table 3-4. **Keystroke Equivalents of Menu Commands**

Command	Keystroke Equivalent	Menu
New	Ctrl-W	File
Open...	Ctrl-O	File
Merge...	Ctrl-N	File
Close	Ctrl-C	File
Save	Ctrl-V	File
Save as...	Ctrl-M	File
Abandon	Ctrl-A	File
Quit	Ctrl-Q	File
Output...	Alt-O	Global
Protection...	Alt-S	Global
Save Preferences	Alt-R	Global
Hide/Show Parts	Alt-P	Global
Hide/Show Tools	Alt-H	Global
Cut	Alt-C	Edit
Copy	Alt-Y	Edit
Paste	Alt-A	Edit
Delete	Alt-D	Edit
Info...	Alt-I	Options
Name...	Alt-N	Options
Type...	Alt-T	Options
Load...	Alt-L	Options
Sort Children...	Alt-F	Hierarchy
Unhide Children	Alt-U	Hierarchy

End of Section 3

Section 4

RSCREATE and Other Technical Information

USING RSCREATE

RSCREATE is a C language utility program that creates a resource file. It is primarily intended for two purposes:

- To create a resource file from a hand-edited file.
- To port resource files between microprocessor environments.

RSCREATE uses a .RSH file (described next) as an include file. To generate a new .RSC file with RSCREATE, include the .RSH file in RSCREATE.C and compile, link, and run RSCREATE.

.RSH File

The .RSH file, which is an ASCII file you can edit with a text editor or word processor, is one of the optional GEM RCS output files. To make GEM RCS create a .RSH file, choose the **Output...** command from the Global Menu, and click on the box labeled "*.RSH" in the output file dialog. You can make the .RSH file one of the default set by choosing the **Save Preferences** command from the Global Menu. See the description of the Global Menu in Section 3.

GEM RCS only produces output files as part of the process of saving a resource file. For that reason, if you open a resource file solely to produce a .RSH file (in other words, without editing the file), you must use the **Save As...** command and then enter the file's current name in the Item Selector. Because the resource file has not been edited, closing the file with the **Close** command or the close box does not save the file and does not create a .RSH file.

Hand-Edited Resource Files

To hand-edit a resource file, generate a .RSH file and make your changes in the .RSH file with a word processor or text editor. Before you hand-edit a resource file, however, be sure you are familiar with the material in the GEM AES Reference Guide describing the Object Library and the Form Library.

When you hand-edit a .RSH file, remember the following:

- When adding a new object, TEDINFO, etc., insert it at the end of the current entries.

- If you insert new objects into existing trees, update the tree base definitions.

- GEM RCS expects object trees in the following order: root first, followed by its children, left to right, with the rule applied recursively. Make sure you enter any hand-built trees in this same order.

- If you are creating a resource file with GEM RCS and you expect to hand-edit the file later, don't enter any object names while you're in GEM RCS. Wait until you've edited the .RSH file and have run it through RSCREATE. Then read the new .RSC file back into GEM RCS and enter the object names.

  If you enter object names in your first pass through GEM RCS, the .H and .DFN files are linked to the object numbers assigned in that first pass. RSCREATE changes these object numbers, and GEM RCS will not be able to use the .DFN file for any subsequent work on the resource file.

- GEM RCS automatically sets the LASTOB flag in the OBJECT structure of the last object in each tree. (See the descriptions of the OBJECT structure and object flags in Section 6 of the GEM AES Reference Guide.) If you hand-edit a tree, make sure this flag is set for the last object in the tree. If you don't, your application can crash at run-time.

Porting Resource Files

To port a resource file from one environment to another, do the following:

1. Move the .RSH file to the new environment.
2. Add a header to make the .RSH file compatible with the target environment. If you are porting to a 68K environment, the header is commented out in RSCREATE.
3. Enter the target-format .RSH as an include file in RSCREATE.
4. Compile, link, and execute RSCREATE on the target environment.

Compiler Notes

Compile RSCREATE with Lattice™ C or another full language implementation.

Some C compilers (including Lattice C) fold duplicate strings together when you compile RSCREATE. This can present problems at run-time if the strings are to be edited. In that case, run the resource file through GEM RCS again to resolve the duplicate strings.

CHAINING RESOURCES

The GEM AES enforces an absolute limit of 64K for a resource file. If you need a larger resource, you can chain two resource files together. However, to use the second file in the chain, you must first clear from the AES any references to the first resource file, and you must clear the first file from memory by making a RSRC_FREE call before you load the second file.

UNKNOWN OBJECT TREE TYPES

If you load a resource file and do not have its .DFN file, GEM RCS displays each tree as an UNKNOWN (represented by a question mark) or an ALERT. Before you can work on the file, you must convert the unknown trees to one of the known types. You should also give each tree a name for the .DFN file you will create when you save the file.

To assign types and names to the trees, do the following:

1. Click on the first UNKNOWN (TREE1) to select it.
2. Display the Options Menu, choose the **Type...** command, and select the DIALOG option. This replaces the question mark icon with the DIALOG icon.
3. Double-click on the TREE1 icon. This opens the tree, and you can then see what it is.

If the tree really is a dialog, do the following:

1. Close the tree to return to the root level.
2. Choose the **Name...** command from the Options Menu.
3. Give the tree a name.

If the tree is not a dialog, do the following:

1. Close the tree to return to the root level.
2. Choose the **Type...** command from the Options Menu.
3. Select the correct type for the tree.
4. Choose the **Name...** command from the Options Menu.
5. Give the tree a name.

For the ALERT trees, you need only open them to see what their messages are and then use the **Name...** command to give them names.

Using the **Type...** command from the Options Menu, you can change the type designation of any tree or object. Remember that MENU and ALERT trees are quite restrictive; they accept only specific objects that are for the most part not compatible with other object trees. You can change a tree's type from a restrictive type to a less restrictive type (for example, from ALERT to DIALOG), but you should not try to go from less restrictive to more restrictive. To do so can produce unpredictable results.

ARRANGING AND ALIGNING OBJECTS

Especially in a dialog, you might want to have several objects arranged in a particular way. For example, you might want three or four text strings to appear in a right-aligned column. To do this, you can first arrange them inside a box, as follows:

1. Drag a box from the parts box to the view window. Make the box the size you want.
2. Add the STRING objects to the box and edit them.
3. Select all of the strings in the box using the Shift-click technique.
4. Display the alignment pop-up menu and choose the alignment you want. (As long as the strings are inside their parent box, you can also move them as a group.)
5. Select the parent box and then choose the **Remove Parent** command from the Hierarchy Menu. Removing the parent box reduces the size of both the tree and the resource file.

CREATING A LIBRARY FILE

If you use certain icons, bit images, or subtrees frequently, you can collect them in a "library" resource file. Using the **Merge...** command (File Menu), you can read this library file into your current resource file. You can then use the trash can or clipboard to cut any extraneous objects from the resource file.

End of Section 4

Index